for first mailings of April Fool's
Day. Those who have made submissions
to this and previous software distri-
butions will be receiving release
forms which must be signed and
returned before we distribute the
tape.

STAFF NEWS

We are happy to report that the two
part-time employees of the associa-
tion authorized by the Board of
Directors are aboard. As a conse-
quence we hope to be able to process
your letters and phone calls with
greater promptness than in the past.
Susan Roth and Lorrin Wong have
joined Armand Gazes and Mel Ferentz,
who continue to serve as volunteers.
While we continue to prefer written
inquiries, we will try to handle
telephone calls. Please avoid asking
for a specific individual when you
call, if possible. If you tell the
person answering the nature of your
call, it will be handled by the
appropriate available person. The
number is 212-360-1182. If we are
inundated with telephone calls, we
may have to restrict this service to
Institutional Members but, assuming
reasonableness on both sides, we
would hope that wouldn't be neces-
sary.

EXCHANGE ARRANGEMENTS

The Usenix Association, while accept-
ing members world-wide, has no desire
to compete with similar organizations
in other countries. In fact, we are
most anxious to enter into coopera-
tive arrangements with Unix users'
group abroad. We have spoken with
the groups in Great Britain and Aus-
tralia and look forward to formal
agreements with these and other
groups that would give each group the
right to reproduce for its membership
tapes and newsletters of the others.
Canada is a special case in that the

people we spoke with in Toronto
seemed less interested in a seperate
organization than in the problems of
moving tapes across the border. If
that is the case, we would be happy
to contract with some Canadian licen-
see to reproduce and mail the distri-
bution tapes within Canada.

NOTICES

This document may contain information
covered by one or more licenses,
copyrights, and non-disclosure agree-
ments. Permission to copy without fee
all or part of this material is
granted to Institutional Members of
the Usenix group provided that copies
are made for internal use at the
member campus or plant site. To copy
otherwise, or to republish, requires
specific permission.

Editorial material, payments, soft-
ware submission, subscription re-
quests, and address changes should
be addressed to:

Usenix Association
Box 8
The Rockefeller University
1230 York Avenue
New York, New York 10021

UKUUG

UNITED KINGDOM UNIX USER GROUP

COMMITTEE
Chairman : Alan Mason, Heriot-Watt University
Secretary: Bruce Anderson, University of Essex
Member(s): Peter Collinson, University of Kent

R.A. Mason,
Department of Electrical and
Electronic Engineering,
Heriot-Watt University,
31-35 Grassmarket,
EDINBURGH. EH1 2HT

Tel: 031 225 8432 Ext: 155

8th January, 1980.

Dr. M. Ferentz,
Box 8,
The Rockefeller University,
1230 York Avenue,
New York, NY 10021,
U.S.A.

Dear Mel,

    With reference to our recent telephone conversation I am pleased to
confirm the tentative arrangements made. We will send copies of all Newsletters
produced and of any software generated direct to yourselves (at no charge) as
our part of the reciprocal arrangement. This is by far the most sensible method
as it allows us to disemenate only that information which is pertinent to our
own communities. On the possibility of operating a similar scheme, I am contacting
the Australian group directly but I would be glad if you would inform any other
major groups (Canada?) of my eagerness to make contact.

    The U.K. group, representing 48 local sites and 30 other subscribing
sites meets quarterly and although Newsletters are at present bi-annual it is
hoped to step up production. A Newsletter will be produced following each
meeting and as well as containing a report on that meeting, will contain information
about the forthcoming meeting.

    Local UNIX user groups (LUUG's) are being established here so that site
representatives may have more frequent contact (monthly). These groups look after
their own internal software distribution and software policy. Thus reducing the
number of centres to which we ourselves have to distribute.

    As the number of installations increases we are being forced to give
ourselves a more formal structure. Although I realise that we are constrained
by different legal and financial systems, it would be of great interest to me
if you could send information on the final structure that USENIX took on
(Byelaws etc.). Could you also tell me how you differentiate between academic
and commercial sites, particularly with respect to software distribution.

            Yours sincerely,

            R.A. Mason

THE UNIX USERS' GROUP MEETING IN SUMMARY

- or -

Reflections of a Scribe

UNIX Conference
Medical Sciences Building
University of Toronto
June 20-23, 1979

This report is a summary of the exciting things people had to say at the Toronto
conference. As I was the only person taking notes for this purpose (although I was
helped out by copies of speakers' transparencies), the notes inevitably reflect my own
personal biases and knowledge (or lack of such). I have tried to convey enough infor-
mation for those who did not attend the conference to get a good idea of the principal
features and availability of the systems and projects the speakers discussed. At the
same time, I hope those who attended can use this report to remind them of all the
thrilling and not-so-thrilling things they heard.

I do not guarantee that what I report was actually said. If you want to be SURE, check
with the speaker in question. My apologies to everyone whom I have misquoted.

My thanks to Sandra Wright for taking the notes of the last session (Saturday morning).
Thanks also to Jim DelGrande, Dave Galloway, Gerardo Lastra, David Miller, Sanand Pa-
tel, Rob Pike, Bill Reeves, Henry Spencer, Dr. Martin Taylor and my wife Simone for
proofreading and helpful comments on the draft.

June 29, 1979

David Sherman
Computer Systems Research Group
University of Toronto
Toronto, Canada   M5S 1A1

Table of Contents

Index by Speaker

WEDNESDAY MORNING
Session 1: LANGUAGES ON UNIX
Chair: Ron Baecker, University of Toronto

Speaker 1, at 9:15 a.m.   **Pascal and EM-1**

Prof. Andrew S. Tanenbaum
Vrije Universiteit — Wiskundle Seminarium
Box 7161
1007 MC Amsterdam, The Netherlands

Andy Tanenbaum gave a detailed talk on the UNIX Pascal system developed at the Vrije Universiteit by Johan Stevenson, Hans van Staveren and himself. Their aim has been to make the compiler and support software as portable as possible, with transport to various mini's and micro's in mind. They feel that "C is not the beginning and end of the world," and that Pascal's attractiveness lies in its simplicity and readability.

The Pascal language implemented is the complete language, and is compatible with the "British Standard," which has been proposed as an international (ISO) standard. There are also a few extensions, such as assertions, mark/release, external procedures, and UNIX-style zero-terminated strings. It is also possible to create libraries of separately compiled Pascal, C, or assembly language procedures that can be called from Pascal programs.

There are also many debugging and performance facilities, such as warning messages for unused or undefined variables; run time error messages giving source line number; counts of source line executions; and procedure entry/exit tracing.

The system can be used to produce either compiled or interpreted code: "pc prog.p" will produce interpreted code, and "pc -C prog.p" compiled code. In both cases the compiler produces code for an abstract stack machine called EM-1, described in *CACM*, March 1978. If the -C flag is given, the EM-1 code is passed through a machine-independent optimizer, and then translated to the UNIX assembly language to be assembled and loaded by as and ld. If the -C flag is absent, the EM-1 code is assembled to EM-1 binary for subsequent interpretation. There are 64 versions of the interpreter corresponding to 6 independent debugging options. The system automatically selects the correct version from a directory, if it is found, or creates and saves a new one dynamically, if need be.

The optimizer's capabilities include constant folding (e.g. mapping 10+4 into 14), using special instructions (e.g. increment for i:=i+1), strength reduction (e.g. shifting for multiplication), reordering (e.g. -k/6 becomes k/-6), and many more.

The distribution is available as part of the 1979 conference distribution. It contains all the sources to the compiler proper, the optimizer, the EM-1 to PDP-11 translater, and the interpreter, as well as various support programs. There are also libraries, including one containing all the UNIX system calls, so it is possible to call open, seek, fork, exec, etc. from Pascal programs. The tape also contains versions of as and ld capable of handling files larger than 64K bytes. Separate I and D space is useful, but not required.

Anyone who installs the system should write to Andy to be put on the mailing list for bug reports (the first of which was available at the conference). As soon as the version 7 Pascal system becomes available (in the fall) both it and the version 6 one will be distributed. To avoid sending tapes across the ocean, the Vrije Universiteit has made an arrangement with the Pascal group of Intermetrics, Inc. (see below), for those who do not have the conference distribution.

Speaker 2, at 10:00   **Pascal and EM-1**

Garry Fostel Intermetrics Incorporated 701 Concord Avenue Cambridge, Mass. 02138

Garry Fostel discussed the Pascal compiler described above. He said it is exceptionally well put together and documented. It is available in the United States from Intermetrics for a $50.00 handling charge.

It runs on standard UNIX V6, and there are certain assumptions about file names near the root. There may be a version for PWB soon.

coffee

Speaker 3, at 10:45 a.m.   **Euclid**

James R. Cordy
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1

The Toronto Euclid Compiler Project is a joint project of I.P. Sharp Associates (better known for their APL services) and CSRG, funded by the U.S. Defense Advanced Research Projects Agency (DARPA) and the Canadian Department of National Defence (DND). It is a follow-up on an attempted implementation by the Systems Development Corporation, which failed in an earlier attempt to produce a Euclid compiler.

Euclid is the so-called "language of secure systems", where secure means *provably* secure. The Euclid language is roughly based on Pascal but has many extensions. It is *very* strongly typed; it has explicit visibility and scope control, unlike Pascal; it prohibits aliasing, or having two identifiers for a single object. This is necessary for program verification and extends even to assuring that no two actual parameters to a routine address the same element of an array.

The heart of the language is the *module*. A module operates as a single mechanism which manages a set of data structures internally and "exports" only a clearly defined set of externally visible operations and data. The program external to the module can invoke any of the exported operations of the module and can access the exported data, but knows nothing of the internal implementation of the module. Thus, the module looks like a "black box" which provides certain services and information through its exported operations and data. It is a self-contained "package" of procedures, variables and constants.

Types can be parameterized in Euclid. A parameterized type definition defines a template for a "family" of types defined by different parameter values. In its simplest form, a parameterized type can define a set of range types whose upper and/or lower bound depends on a parameter value. The full power of parameterized types is realized in the *parameterized module*, which can define a whole family of similar mechanisms.

Constants can be "manifest" – like constants in most other languages – or "non-manifest". The value of a non-manifest constant can be a run-time determined variable expression, but is "constant" in the sense that, once set, the value remains constant in the scope.

Euclid provides pre- and post-assertions for routines, which enable the specifications and assumptions of a routine to be encoded and checked at run-time. Modules may have "invariant" assertions which specify invariant conditions of the module. These also can be checked at run-time when entering and exiting routines exported from the module.

Pointers in Euclid are restricted to a "zone" of storage maintained either by the system or by a user module, and to a particular type of object (which may be paramaterized).

The project schedule so far has been:

Sept. 1977 – Transliterator of Small Euclid to C – so as to bootstrap by writing the entire Euclid compiler in Euclid.
June 1979 – Translator – compiler for a large subset of Euclid.
Fall 1979 – Compiler should be finished. It will cover most of Euclid, and will include all verification features. Implementation of an automatic verifier is not within the scope of the compiler project.

The compiler does not run without separate I and D space. There are 4 machine-independent and 2 machine-dependent passes to the compiler. Performance: Object code produced is generally as good as or better than C in space and time. Because there is so much modularization in Euclid, procedure calling has been carefully optimized. A call to a parameterless procedure with no local variables costs only two instructions (JSR, RTS). The source code for the Euclid Translator is about 60,000 lines. To compile the compiler takes about six hours (two to three hours for the largest pass; there are six passes) on a relatively quiet 11/50. No attempt has been made to tune the compiler for speed.

Euclid is not available publicly yet, although the current version is being delivered to DND (Canada) and DARPA. When the project is complete, it is hoped that IPSA will distribute it commercially (with full maintenance) and that it will be available (without support) to educational and research institutions from CSRG. If you would like to be placed on the mailing list for information about distribution of the Euclid Compiler when it becomes available, send your name to Jim Cordy (address above).

Speaker 4, at 11:35   **Path Pascal**

Richard Bolocca
University of Illinois
Champaign, Illinois

Path Pascal is a superset of Pascal P. It has additional features for concurrency; for data encapsulation; and for self compilation. It is used for the teaching of systems programming at the University of Illinois. It runs on a Cyber and a PDP-11, and may be used for real-time programming by the NASA space shuttle program. Path Pascal was developed by Roy Campbell at the University of Illinois.

Concurrency: features are the *process*, which is like a procedure but "goes its own way"; and interrupt processes. Data encapsulation: there is a new type, called an "ob-

ject", much like a module in Euclid or Modula; processes can be created dynamically with this.

The compiler is self-compiling and takes two passes to reach assembler code. The compiler source is about 4K lines for each pass; the binary takes 48K and 32K bytes for the two passes. The second pass compiles itself in about 4-5 minutes. The compiler needs separated I and D space to run, or non-separated I and D with "segmentation" overlays.

The language currently runs on UNIX but can also run standalone on an LSI-11. It is highly portable: in existence now are compilers for the Z80, all PDP-11's, and the Series/1. Being developed is one for the Prime 500 and similar models.

Path Pascal will be available by the fall for a nominal (tape-handling) fee.

Speaker 5, at 11:45   **Languages for the VAX-11/780**

Bill Joy
University of California at Berkeley
984 Riley Drive
Albany, California 94706

Bill discussed the various languages and near-languages running under UNIX on the VAX at Berkeley:

1) *Pascal.* The Berkeley Pascal is well known. Like the front end of the portable C compiler, its front end is relatively machine independent. The program pi (Pascal interpreter) is being modified to become pc, which will output code to be compiled by either pcc on an 11 (generating PDP-11 assembler files) or pcc on the VAX (generating VAX assembler).

2) *LISP.* The LISP compiler, a.k.a. "Franz Lisp", is written in C. A simple interpreter is running, but the system so far can run only 2/5 of MACSYMA (two of five parts of the MACSYMA compiler which take up 400K and 350K bytes respectively have been run together on a ½-megabyte machine). Work is now proceeding on a "Byte LISP" for compactness, and a compiler.

3) *APL.* Work is being done on moving Ken Thompson's APL to the VAX. The advantage of running it on the VAX is that one can get around the 65K workspace limitation (with paging) and run real APL.

4) *Modula.* The compiler, originally BCPL code from York (England), was rewritten line for line to compile as C, and has been taken over to the VAX. It does not yet compile VAX code.

5) *Rigel.* This relational database language was described in the proceedings of the last SIGMOD conference (at Boston).

6) *MACSYMA.* This system is supported by the LISP system described above.

The following are languages which the Berkeley people are thinking about and/or interested in:

7) *3.* A modified version of FORTH. Since FORTH is proprietary, no more need be said...

8) *Algol 68*. They would like to move this to the VAX.

9) *Snobol/Spitbol*. It would be especially nice to get a Spitbol up.

10) *STAPLE*. This is being worked on. It is a structured applications and programming language for modification of system programs. It's based on C, PL/1, Pascal....

None of the above languages is available officially (i.e., nothing is packaged) except for the Pascal interpreter.

The following can only marginally be considered languages...:

11) *Termcap*. Makes the screen editor terminal-independent.

12) *C Shell*. A shell with its own command language, much like C.

Speaker 6, at 12:15

C Compiler for the Z80

Keith Davis
Teletype Corporation
4 Mayflower
Vernon Hills, Illinois 60061
(213) 982-3819

The C compiler for the Zilog Corp. Z80 is a cross-compiler which runs under PWB. Developed by Interactive Systems, it features everything in the C compiler except floating point. It generates specific Z80 code, which is 42% larger than PDP-11 code because of instruction inefficiency.

The compiler is available, along with a debugging package (like cdb), for those within the Bell system from Keith Davis of Teletype. Internal charge: $5,000. Outside the Bell system, it's available from Interactive Systems (Keith can point you at the right people).

LUNCH

- 8 -

WEDNESDAY AFTERNOON
Session 2: THE UNIX OPERATING SYSTEM
Chair: Bill Reeves, University of Toronto

Speaker 7, at 2:00 p.m.

UNIX Version 7

Brian Kernighan
Bell Labs
Murray Hill, New Jersey 07974

A lot of the Version 7 changes were covered in the BSTJ issue on UNIX. The important new things are: the shell; huge files (2^{30} bytes); portability; the portable C compiler; lint (C program checker); stdio package; Fortran 77 compiler; make; lex (lexical analysis phase of yacc); awk (pattern scanning and processing language); sed (stream editor); learn (computer-aided instruction about UNIX); adb (a "complex but indispensable" C debugger); uucp (UNIX-to-UNIX communications handling).

The new shell is much more oriented to programming than previously. Gone is the old "goto"; in its place are "for ... do ... done ", "case ... esac", "if ... then ... else ... fi", "while", "until", and a "trap" command to handle interrupts. Also new are shell variables (including special ones for home directory, mail file and bin path search).

Changes in C for portability are all described in the C book; in addition, structure assignment has now been implemented, and there is an "enumerated type" feature.

Lint is a program checker that goes right through a program (including multi-file programs) and checks for type violations, portability problems, probable errors, and bad style that may be evidence of error.

Make is a command to compile according to instructions in a "makefile". It knows about a lot of things by default, such as that "cc -c zork.c" will produce "zork.o". Its only drawback it that it does not yet handle ar libraries.

F77 compiles the complete Fortran 77 language, with a few extensions. It generates the same intermediate code as C, so UNIX I/O is accessible. Also on the distribution is struct, a program to convert standard unintelligible Fortran to Ratfor.

Awk is a pattern scanning and processing language in which programs are typically 1 or 2 lines. Initialization and declarations are not used -- awk decides what an item is by how it is accessed and used. It is useful for such tasks as switching two fields within a file, or adding up the contents of a particular field.

Learn interprets a script to teach UNIX. Scripts are available for teaching UNIX file handling, ed, C (not a very good tutorial, according to Brian, who wrote it), and the -ms macro package (the Bell people don't think anyone should use nroff directly so there's none for nroff). The lessons are streamed and tune themselves to the competence of the student. Brian mentioned that writing scripts for these things is extremely hard.

Tar, the new tape program, was discussed by Tom London (below, but the information belongs here). Files sent to tape with tp still work. All code is written on the tape in ASCII, so any system can read it. You can block the tape with huge blocks. Updates on the end of the tape are allowed, and extracts are done on the last instance of a file. Directories are built for you when you extract files.

- 9 -

coffee

Speaker 8, at 3:30   **VAX UNIX 32V**

Tom London
Bell Labs
Murray Hill, New Jersey 07974

UNIX 32V is UNIX on a VAX-11/780. Tom described in some detail the mysteries of VAX
addressing, with its 31½ bit address space. Presently running on the VAX is a full UNIX
V7, with the same file system, shell, commands, language and system interface as the
PDP-11. There are just a few incompatabilities: the VAX word size is 32 bits, so pro-
grams which know that a word is two bytes are wrong and ints are aligned at 4-byte in-
stead of 2-byte boundaries; long ints are encoded in the reverse order from the PDP-11;
traps are handled differently; pointers cannot conveniently be assigned the value -1;
external variables declared contiguously won't necessarily be contiguous in memory;
and so on.

The VAX, with its block-move instruction, performs faster than the PDP-11 in tasks
which involve moving large chunks of data at once (although it does not quite live up to
the manufacturer's billing of being consistently twice as fast). Compiling a C program
on the VAX is faster than on the 11/70, even though the **pcc** being used is Steve
Johnson's, which is not as finely tuned as Dennis Ritchie's.

UNIX 32V has partial swapping -- pages can be scattered anywhere in memory, and only
part of a program will be swapped out to make space for another. These improvements
reduced execution times by a factor of 2/3. Demand paging is not yet available, but is
being worked on at Berkeley (see Bill Joy's talk below).

In summary, the VAX is well worth using as a UNIX machine. You get a full V7 on a fas-
ter machine, a larger address space, and the capability of handling huge programs.
UNIX 32V is now available from Western Electric (see Speaker #12 for details).

Speaker 9, at 4:30   **VAX UNIX - A User Comments**

Prof. Ed Desautels
Computer Sciences Department
Wisconsin University
1210 W. Dayton
Madison, Wisconsin 53706

The VAX at Wisconsin is now running UNIX V7. It is part of a network of several PDP-11's
and LSI-11's, hooked up to each other in various ways. The VAX originally ran VMS (with
auto-reboot); the VMS Pascal being developed at University of Washington (Seattle) was
tested at Wisconsin.

VAX UNIX was found to be easy to install, although it is "not yet robust". Projects
underway on VAX UNIX include: an intelligent mass storage system (Dave DeWitt, Tony
Klug); compilers and operating systems for computer networks (Raphael Finkel, Marvin
Solomon); a data base system for AI (Larry Travis); a Computer Science network for
Telenet (Larry Landweber; Ed Desautels); connection with a large array of micros for
perception research (Len Uhr); new architecture and systems work (Ray Moore, Bob

Cook, Ray Bryant and Goodman); and facilitation of access to computing for the visual-
ly impaired, along with work on the Teletext/Viewdata systems (Ed Desautels).

Speaker 10, at 4:45   **KSOS -- Secure UNIX**

Howard S. Weiss
U.S. Department of Defense
9800 Savage Road, Room 171
Fort Meade, Maryland 20755

KSOS stands for Kernelized Secure Operating System. It started as an ARPA project in
1977 and is being coded in Modula. (Euclid was the original choice but the Euclid pro-
ject was not ready in time.) Written by Ford Aerospace at Palo Alto, it should be com-
pleted this winter.

KSOS is an operating system in itself. The "UNIX" that runs on it is a UNIX emulator.
Other emulators may be written, should the demand warrant it. The UNIX emulator is
committed to performing at no worse than one-half the speed of UNIX V6.

KSOS will be available to those within the U.S. federal government system. It is un-
known yet whether it will be made publicly available.

Speaker 11, at 5:00   **What's cooking on the Berkeley VAX**

Bill Joy (again)
University of California at Berkeley

Projects underway on the VAX called "Ernie" (½ megabyte) include:

1) paging (Ozalp Babaoglu, Juan Porcar) -- see below
2) microcode work (Dave Paterson, Richard Tuck)
3) a new floating point box (George Taylor, V.V. Kahan)
4) a floppy-disk driver (Richard Tuck)
5) swapping to UNIBUS disks (Eric Allman, Bob Kvidle)
6) LNI interface for networking (Ken Birman, Larry Rowe)

Because the hour was late, Bill restricted himself to talking about paging. The goal of
the pager, called PUNIVAX, is to support large programs, with mostly large text seg-
ments (up to 6 megabytes of text). There were some clear architectural problems to
overcome, in that the VAX has no reference bits and has very small pages (512 bytes).
The paging system is well on its way now, however.

THURSDAY MORNING
Session 3: Items of Interest to Users
Chair: Mel Ferentz, The Rockefeller University

Speaker 12, at 9:00 a.m.　　　　　　　　　　　　　　**News from Western Electric**

Al Arms
Western Electric Company, Inc.
Box 25000
Greensboro, North Carolina 27420

Al reminded us all that Western Electric sells a UNIX license "as is", with no warranty of
any sort and no maintenance or service. He announced that UNIX Version 7 and VAX
UNIX 32V are on the market. Western Electric's commercial rates are now:

System	1st CPU	Add. CPU	Binary
Mini UNIX	$12,000	$4,000	
UNIX V6	20,000	6,700	8,400
PWB 1.0	30,000	10,000	12,000
V7	28,000	9,400	11,700
32V	40,000	15,000	18,000

For users who already have a UNIX license, V7 may be obtained for $12,000, and $4,000
for additional CPU's. No such discount is available for 32V. Educational institutions
may get an "administrative" license, for internal business uses, at one-third of the
commercial rates. The educational research price is $300 for V7 or for VAX 32V; $230
for V6 tape and manuals.

PWB Version 2.0, which has the V7 file system and is fully compatible with V7, was just
released internally within Bell. It will not be available publicly for some time.

MERT will not be released (i.e., it officially does not exist any more). V7 will probably
not be released for the Interdata 32-bit machine (8/32). A point of interest: there are
currently about 800 V6 licenses, of which 250 are commercial.

Speaker 13, at 9:45　　　　　　　　　　　　　　　　　　　　**USENIX**

Lou Katz　　　　　　　　　　　　Melvin Ferentz
Columbia University　　　　　　　The Rockefeller University
New York, NY　　　　　　　　　　1230 York Avenue
　　　　　　　　　　　　　　　　New York, NY 10028

At the users' meeting in New York last year, a committee of five was nominated to in-
vestigate the setting up of a formal UNIX Users' organization to handle such matters as
tape distribution, newsletter publication and conference planning.

On June 20, 1979, an association was formed called USENIX. It has a legal existence in
the State of New York. Its Board of Directors initially consists of: Lou Katz, President;
Lew Law, Vice-President; Armand Gozes, Secretary; Mel Ferentz, Treasurer; Mars Gralia
and Peter Weiner, members of the Board. USENIX invites institutions and individuals to
join; it has no exclusive "rights" over anything and no-one is obliged to join.

If USENIX becomes successfully operational, it will set up in business at the Rockefeller
University, handling the affairs described above.

Membership in USENIX will fall into one of four classes: (a) voting; (b) individual; (c)
public individual; (d) non-voting institute. Voting members will be institutes with a
UNIX license who pay $300 for each vote, to a maximum of the number of CPU's for
which they hold licenses. The educational-institution rate will be $100 instead of $300.
Voting members will delegate individuals to exercise their institution's vote. Individu-
als will be permitted to join for $12 and receive the newsletter, which will be produced
at least 10 times per year. Individuals who work for or are associated with a UNIX
license holder and are therefore bound by the license's non-disclosure clauses will hold
regular individual membership and be entitled to receive information about
proprietary parts of the system. Those not bound by non-disclosure may hold an "out-
side" membership. Non-voting institutional membership will be available to those in
the Bell system (who are of course not "licensed").

USENIX will hire a full-time employee to handle the newsletter, tape distribution and
the maintenance of a data base on the UNIX. The facilities at the Rockefeller University
will be available for this purpose.

The bylaws and an invitation to join will be distributed to all parties thought to be in-
terested. Election of a new slate of directors will be held by mail, probably around the
December. The association may be contacted by mail at:

　　USENIX Association
　　Box 8
　　The Rockefeller University
　　1230 York Avenue
　　New York, NY 10021

Lou announced that the Association is currently asking for volunteers to serve on four
committees: Agenda for the Winter Conference at Boulder; Site Selection for next
year's Summer conference; distribution tape format; and nominations for office for De-
cember elections (but see Speaker #37, below).

Speaker 14, at 10:30　　　　　　　　　　　　　　　　　　**Winter Conference**

John Donnelly
National Center for Atmospheric Research
Boulder, Colorado

A brief announcement that the Winter Conference will be held January 29 - February 2
(Tuesday through Saturday) at a convention centre in Boulder. All users are en-
couraged to bring their /dev/ski drivers.

coffee

Speaker 15, at 11:00   **UNIX without a UNIX license**

Mark Krieger
Whitesmiths Limited
127 East 59th Street
New York, NY 10011
(212) 799-1200

Whitesmiths have started from a UNIX binary license and created their own C compiler for the PDP-11. The compiler produces either UNIX or Macro-11 assembler. It supports the full Version 7 compiler as defined in the Kernighan and Ritchie book. (Extensions to that syntax mentioned by Brian Kernighan in his talk, such as structure assignment, will be provided for if these become standard.) The compiler runs under any of UNIX, RSX-11M, RT-11, IAS and RSTS.

A portable library comes with the C compiler. It features alloc and free; char, line and formatted I/O; and string functions. Stdio is not followed too closely.

A PDP-11 machine library provides certain functions that the hardware can't do on some machines, such as floating-point arithmetic.

Currently available is a C compiler for the 8080. It runs on any PDP-11 and takes the same V7 C as input. It generates an intermediate assembly language called "a-natural". The translator from a-natural to ISIS-II or CP/M microsoft assembly code is the fourth pass of the compiler. It generates code only 50% larger than the corresponding PDP-11 code (not bad considering the instruction inefficiencies in the 8080). The machine library for the 8080, written by Bill Plauger, supports all 16-, 32- and 64-bit operations supported by C.

By July 1979 Whitesmiths will be distributing a full "a-natural" assembler, loader and librarian for the 8080. It will run on the PDP-11 or the 8080 on any of the operating systems named above.

By the fall the IDRIS operating system (named after the Persian deity over toolsmiths) will be on the market. It will look exactly like UNIX and will be runnable on an LSI-11. Memory management should be available by the early winter.

By early 1980, a native C compiler for the VAX and C compiler for a 16-bit micro should be available.

Mark went into some detail about the features of the "a-natural" assembly language. The C compiler currently available costs $5,000 (including source). Without the source, it's $500.

Speaker 16, at 11:55   **What's on the Berkeley tape**

Bill Joy (one more time!)
University of California at Berkeley

The tape includes: Berkeley Pascal version 1.2; ex (display editor which requires separate I and D space) version 2; C shell (runs on either V6 or V7, good while converting); -me macro package (faster than -ms); a new mail program; the Berkeley networking (for machines hooked up by back-to-back DZ-11's; mods to stdio for simultaneous read/write; miscellaneous utilities; and a V7 simulator library (system dependen-

cies can all be left in this lib and changed at once).

For each item one will find: full source; V6 binaries; V7 makefiles; all manual sections and documents; and a Versatec copy of the printed documentation. Everything except Pascal runs on both VAX UNIX and V7. Everything runs on 11/40's and 11/34's.

To get the tape you need $60, a copy of your UNIX license, and a V7 or Photolypesetter license for the -me package. You will get a 1200-foot tape at 800 bpi.

Speaker 17, at 12:10   **YASL**

David Lilienfeld
Johns Hopkins University
34th & Charles Streets
Baltimore, Maryland 21218

YASL -- Yet Another Statistical Language -- was designed because there was a need for a statistical package on a small machine. The reasons existing packages (such as BMD or SPSS) could not be used were (a) they are clumsy; (b) you "can't tell them what you want them to do"; and (c) you need a large machine, such as a DEC-10 or a CDC 6600, to run them because of the memory they take up.

The writers of YASL have emphasized structure in their language, which they have tried to make a superset of C. Certain features useful for matrix statistics have been designed, such as case ranges (e.g., case 'a'-'z':), and declarations for various types of matrices. Matrices can be dynamic and/or virtual. Virtual matrices are necessary to run large regression analyses and the like on a machine with raltively little memory.

The compiler is currently being written. It is the aim of the authors to have something running by the end of the summer, and a finalized version by June 1980. At that point it will become generally available (including the source).

LUNCH

THURSDAY AFTERNOON
Session 4: Graphics, Music and Typesetting
Chair: Ron Baecker

Speaker 18, at 1:55 p.m.   **Core Graphics in C**

Dennis Mumaugh
U.S. Department of Defense
9800 Savage Road
Fort Meade, Maryland 29755

Core Graphics is an "Independent Graphics System" designed by the writers of the Core Report (see *ACM Computing Surveys*, December 1978 for a whole issue on Core). It features a set of common subroutines for any language to do all the basic things a graphics language should do. At the bottom level are device-dependent primitives to do such tasks as basic line drawing.

Dennis reported that a Core Graphics for C has been written, although it is not yet complete. It has been checked on the Version 7 C compiler. The only device driver with it is one for a Genesco graphics system (which might not be of help to anyone, since Genesco itself has no standard graphics format). It differs from Core in a few details, such as not having function names longer than 7 characters (the maximum distinguished in C).

This graphics package is available as part of the conference distribution. The Core manual is not on it; that may be obtained from SIGGRAPH on machine-readable tape. This tape has a list of differences between the package and the SIGGRAPH manual. Some test programs and instructions on how to build the driver are supplied.

Speaker 19, at 2:05   **Perception and Information Enhancement**

Martin Tuori
Defence and Civil Institute of Environmental Medicine (DCIEM)
1133 Sheppard Ave. West
Downsview, Ontario, Canada M3M 3B9
(416) 633-4240 x204

Martin Tuori and Dr. Martin Taylor are doing work at DCIEM on human perception. Their aims are to study vision and audition in complex situations, and to develop visual and auditory enhancements which will aid persons working in those situations. Experiments will be conducted to demonstrate the effectiveness of such enchancements. The techniques will be of use with information from various sources, including earth resources satellite imagery.

Martin discussed the "IPAC" file format for storage of picture, sound and other data, and invited interested persons to contact him. The format is hierarchical; the structures inherent in the data are first presented in a header. This will permit programs to interpret data files of varying structure. He would like to discuss the IPAC technique, with a view to working out a more widely accepted standard.

Martin then showed slides, generated with a Dicomed D48 film recorder, including: (a)

photographs of parts of Canada from LANDSAT satellite, and various techniques by which they can be enhanced; (b) images made with *fns*, a program written by Mike Tilson at the University of Toronto to generate coloured patterns based on x-y-r-theta functions and a 256-colour map; and (c) pictures of Mars taken by Viking orbiter and brought from Caltech by Rob Pike.

Speaker 20, at 2:28   **GPAC and veat**

Ron Baecker
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1
(416) 978-8768

There is nothing new to be said about GPAC. It continues to be used as the graphics package at the University of Toronto for building highly interactive systems (such as ones for producing animated film, composing and performing electronic music and visualizing the behaviour of simulation models). It is available and fully documented as part of the Toronto distribution. GPAC was designed and written by Bill Reeves, who also wrote *veat*, a Versatec typesetter-simulator which runs as an output filter to troff. Veat (which was used to generate this document) is available free as part of the University of Toronto distribution #3.

Speaker 21, at 2:30   **NYIT Graphics**

Tom Duff
Computer Graphics Lab
New York Institute of Technology
P.O. Box 170 Old Westbury, NY
(516) 626-0938

The Computer Graphics Lab at NYIT is in the business of making cartoons, ranging from short spurts to (allegedly) full-length features. They don't distribute any of their software. They have lots of hardware: 10 PDP-11's which run UNIX, lots more which don't, a VAX, 18 frame buffers (512 x 512 x 8 bits each), and, as the ultimate peripheral device.... a TV studio. Need we say more?

Tom showed a very entertaining 20-minute videotape about NYIT's graphics.

coffee

Speaker 22, at 4:00   **Graphics for lots of different terminals**

Mike Muuss
Johns Hopkins University
Electrical Engineering Department
Barton Hall
Baltimore, Md. 21218

Mike Muuss described the "Terminal Independent Graphics Package" at Johns Hopkins (funded by the Army Research Office). The goal of the project was to create a package which would let one create a graphical display only once, and use it subsequently on different terminals without re-executing the original "a.out".

The system has a graphics format which lets one add new devices easily. It is also possible to reprocess the display after generation. The display size is exactly the same for a given picture on all devices.

Low level primitives, using the concept of a "virtual pen", have been written for the package: pen up/down, move pen, new form, new origin, set colour/intensity. High-level routines available include rotation, scaling, and line, symbol and number drawing. 3D perspective is under development.

Devices currently supported with the package include the HP2800A series; Tektronix 4008, 4010 and 4014-1; Hewston Instruments "Complot"; Versatec; Diablo 1620 series; and Ramtek Colour Graphics. It takes an afternoon's work to add a new device to the package.

The package is distributed by:

> Dr. Bruce Henriksen
> Ballistics Modelling Division
> Ballistic Research Lab
> United States Army, Aberdeen Proving Ground
> Aberdeen, Maryland 21005

Speaker 23, at 4:10   **Versatec Typesetter Emulator**

G.E. Toth
Johns Hopkins University
Electrical Engineering Dept.
Barton Hall

VERSET was partially developed at Johns Hopkins University but completed and debugged for the commercial firm which markets it. It can run without separate I and D space and fits entirely on an RK disk. The font file takes up 1321 blocks and covers all troff point sizes and widths. VERSET runs as an output filter to troff without any modifications to existing software. It can process a page on the Versatec in 20 seconds.

Verset is available in binary form for $2000 to commercial users, $500 to educational licensees. For more information contact:

> RLG Associates, Inc.
> Suite 16
> 11250 Roger Bacon Drive
> Reston, Virginia 22090
> (703) 471-1106

Speaker 24, at 4:25   **Theories about Office Automation**

David Macfarlane
Bell-Northern Systems Research
522 University Avenue
Toronto, Canada M5G 1W7

David Macfarlane gave a slide show from BNSR called "An Integrated Methodology for Office Automation". In the slide show, the merging of the three fields of computers, communications and office equipment is discussed, and certain design problems in the construction of an "Office Information Communications System" are explored.

The bottom line is that there is a need for an integrated perception for the study, design and implementation of an automated office system. Nothing has been done beyond the thinking stage as yet.

Speaker 25, at 4:50   **Musical Interlude**

Bill Buxton
Structured Sound Synthesis Project
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1

Bill Buxton "conducted" a half-hour concert generated by the SSSP music synthesizer running off an LSI-11 with floppy disks. The entire system is written in C and takes up about 14.5 Kwords of core. The musical patterns were loaded into core at the start of the performance, and Bill "conducted", sitting at a terminal with a bit pad, cursor and function box. The terminal's addressable cursor was used to approximate the graphic environment available on CSRG's 11/45 with a full tablet and vector graphic display (those who came to the demos on Wednesday and Thursday evenings saw the real thing). "Conducting" consisted of varying, as a performer, the volume, timbre, articulation, tempo and other properties of the patterns.

World UNIX Reversi Championship

Thursday, June 21, 7 p.m. - 1 a.m., the World UNIX Champion of Reversi was determined. Reversi, as you all now know, is a board game (also known as Othello) played by two players on a 64-square board.

Eight entries competed, including one written by the organizer, Bill Reeves. Our hardware mastermind (Guy Fedorkow) set us up with nine terminals in the CSRG conference room. Dave Sherman directed the tournament as a 4-round Swiss system (along the lines of a chess tournament), in which players with equal records were paired in each round. We ran three games simultaneously on the 11/50 (i.e., six programs, of which a maximum of three were active at any one time), and one on the 11/45 in competition with the graphics/music demos going on in the lab.

R. Dennis Rockwell of Duke University emerged as the winner with a perfect 4 out of 4, beating Tom Duff in the last round. Bill Reeves finished second, losing only to rdr in the third round.

To ensure that the tournament didn't last all night, a time limit of 15 minutes per program per game was enforced. All games were run with time(1), and any program using more than 900 seconds of user+sys defaulted the game. Fortunately, no-one ran over, although rdr did finish his second game with only 8.5 seconds to spare!

The final scores: rdr (Dennis Rockwell, Duke University) 4 points; bill (Bill Reeves, University of Toronto) 3; td (Tom Duff, NYIT), allan (Allan Wilkes, Princeton), susk (Larry Custead, University of Saskatchewan), and ber (Brian Redman, Bell Labs) all 2 points; mum (team of three from the University of Western Ontario) 1; psl (Peter Langston, New Permanent Wrinkle) 0. Thanks to all competitors and spectators for a great tournament. See you in Boulder?

[Rumour has it that Bill Reeves is planning on holding the first meeting of SIGREV/USENIX in Fiji in the summer of 1980.]

FRIDAY MORNING
Session 6: Interesting projects on interesting UNICES
Chair: Bob Hudyma, University of Toronto

Speaker 26, at 9:15   **Software Register proposal**

Lynn Brock
Computer Corporation of America
1800 Wilson Blvd., Suite 903
Arlington, Virginia 22209
Washington, D.C.

The Computer Corporation of America does UNIX work as a contractor for ARPA in cybernetics technology and the like. CCA has proposed to ARPA, and expects funding by October for, a "UNIX Software Register", to be known as USR or perhaps /usr (groan), an on-line database of information about UNIX-related software. Both commercial and non-commercial materials would be involved, and the list would include programs written in C and compilers for C, regardless of the machine.

The proposed registry would be accessible via an INWATS (800 exchange) free phone number, as well as on the ARPAnet and perhaps TELENET. Users would be able to dial in and access information, enter new information, and update infomation entered by them. The entire database would be maintained by the users, with the administrators merely keeping an eye out for irregularities. USR would not distribute any software; it would be up to those wishing to get it to contact the individual suppliers.

The suggestion was fairly well received by those present. In the light of the creation of the USENIX association, it would be appropriate for these two groups to get together to work out how best to handle the database.

Speaker 27, at 9:30   **Database for a Micro**

John Kornatowski
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1

MRS is the Micro Relational System for information retrieval on UNIX and UNIX-like systems. It is designed for fairly small data bases (up to 10 megabytes). It features: easily-defined data bases, a powerful retrieval facility, simple data entry and interactive data modification. MRS is fully interfaced to UNIX and has been tested in actual applications.

MRS runs on UNIX, Mini-UNIX and LSX, and has been tested by the designers on LSI-11's and PDP-11's of every size. The data can be stored on anything from a floppy to a big disk. The total size of all programs in the package is about 300 blocks, so users can allocate whatever space they have left for data.

John gave us a sample session with MRS in his slides, to be followed by hands-on demos Friday and Saturday at the CSRG UNIX lab. Simple techniques for data entry, retrieval, updating and manipulation were shown on a rated list of Toronto restaurants.

An application being worked on using MRS is the FORM system, designed for handling office forms to be generated and dealt with interactively or automatically. Such operations as filling in a form, copying it, mailing it, discarding it, filing it, attaching it to a dossier and making an audit trail of it can be performed.

The FORM system is still under development. MRS is available for a $200 distribution charge from the MRS Distribution Manager, CSRG (address above). A copy of your UNIX or Mini-UNIX license and a signed University of Toronto Software Release Form must be provided. (See how important forms are?)

Speaker 28, at 9:55   **Mini-UNIX on the LSI-11**

Bob Hudyma
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1

Bob picked a slot in the session he was chairing to tell us how to adapt Mini-UNIX for the LSI-11 and up. The minimum requirements for the system are: the LSI-11; Extended Instruction Set Chip (optional but preferred); Serial Line Board DLI-11; 24 Kwords of memory; and at least two AED or equivalent floppy disks (double density, DMA, 1200 512-byte blocks).

What you can end up with is a single-user system (it could accomodate 2-3 users performing small tasks with a fast disk like the RL01 or RK05); 18 Kwords of user-addressable memory (just enough to hold the C compiler); and a system that can handle practically all the usual UNIX software (nroff, yacc, V6 cc, ed, etc.). Forks are supported, and pipes are simulated via temporary files. The system can compile a 200-line C program to a.out in about 3 minutes. Its execution times compare favourably with LSX (which can, however, run programs about 4 Kwords larger).

Bob now presented the "Instant Do-It-Yourself Mini-UNIX-on-the-LSI Kit":

1) Write a device driver for the AED floppy disk.
2) Replace all references to PS in the system source by subroutine calls. Supply the appropriate routine in mch.s.
3) Eliminate clock references in main.c (these screwed up the LSI for some obscure reason).
4) Modify init.c for single-user only: remove references to checking the console switches.
5) Change the swap strategy to include a "swap flag" so as to prevent swapping while a swap is in progress on those slooooooow floppies.

If you send a copy of your Mini-UNIX license, the source for all this is available with the MRS distribution tape (see Speaker #27 above).

Speaker 29, at 10:15   **Real-time Data Collection and Failure Analysis**

Neil Groundwater
Analytic Disciplines, Inc.
8320 Old Courthouse Road, Suite 300
Vienna, Virginia 22180

Neil spoke about work he did when working for the New York Telephone Company on UNIX starting in 1972. An 11/20 was used for analysing failures in the telephone network with a view to locating areas with recurring problems (which would indicate some defect such as water getting into a phone cable).

Neil described the trouble reporting and analysis phases of the operation. Because of the quantities of data continuously being collected, it was necessary to filter out unimportant information. As a result, an "alerter" was written, which processed masses of data. The alerter would take note of certain problems in the system; any possibly critical problem would be sent to a monitored terminal immediately.

The system is now in use at N.Y. Telephone, collecting data from 96 simultaneous inputs. The data is manipulated by ATOM (Analysis Tool for Maintenance), a graph-oriented series of filters. The system runs now on N.Y. Tel.'s 11/70's, but is not available to those outside the Bell System.

coffee

Speaker 30, at 11:00   **How to get more out of your 11/70**

Dan Gielan
New York Telephone Company
375 Pearl Street
New York, NY 10038

Dan described N.Y. Tel.'s collection of 11/70's, each with 1 megabyte of memory and dial-in/dial-out facilities. Running USG-3 UNIX, they have hit a limit (per system) of 130 simultaneous processes, 300 open files, 320 inodes and 30 buffers. To gain more throughput they have implemented certain changes:
1) a swap device driver that lets the swap space size vary dynamically by overflowing onto another free device when the swap area is full;

2) checks for swapmap and coremap exceptions (so that when the system crashes you at least know why...);
3) the DH-11 driver and tty.c were changed so that in raw mode, the "kill" position in the byte table is used to denote a "message terminator";
4) RP06 drivers were changed by taking an RP04 driver and adding a search capability;
5) disk space assignment was improved;
6) mods were written to speed up the LP-11 driver
In design are a replacement for the RSW04, hanging memory off the MASS BUS; and a "smart" DH-11 driver with a Z80.

Dan had some recommendations as to what hardware should be obtained when funds are limited. In decreasing order of priority, they are:
1) 256 Kbytes of memory with FP11-C floating point box;
2) an independent swap device;
3) another 256 Kbytes of memory (Dan differs from the Bell people on this point);
4) separate controllers for the disks (or hang the TE-16 tape drive off the UNIBUS and free up one more of the 4 high-speed slots on the 11/70);
5) add a "smart" data handler (better multiplexer);
6) if you STILL have money to burn, get another CPU and split up the tasks.

Speaker 31, at 11:20   **Real-time data gathering on UNIX**

Eric Ostrum
Neurological Control Systems Lab
Department of Neurology
Carnegie Mellon University
5508 Walnut
Pittsburgh, Pennsylvania 15232

Eric Ostrum talked about the activities at NCSL in the field of research into brain functions and patterns and neurological diseases. UNIX is used for real-time data gathering of eye movements and gait patterns.

NCSL has developed the following packages: AD (Analog-to-Digital device driver); RA (read and interpret analysis data); PA (plot analysis data); EA (edit and manipulate the data). The system is available for $50 to educational users ($100 to commercial users) on a distribution tape. Separate I and D space is not required to run it.

Speaker 32, at 11:35   **Networking at Purdue**

Bill Croft
Electrical Engineering Dept.
Purdue University
West Lafayette, Indiana 47907

Bill Croft has developed a fairly complex networking facility at Purdue, where they have two 11/70's and two 11/45's. The 70's are connected to each other by DMC-11, as are the 45's. The links are about 1 megabaud bandwidth. A sustained end-to-end throughput of 250 Kbaud is about the same load on the CPU as a disk file copy.

The networking package features arbitrary interconnection: many individual connections are multiplexed through the same physical link rather than spooled. As a result,

a reasonable amount of interaction is available, with simple user commands for virtual terminal connection (con); remote process execution under a "connected shell" (csh); and file/directory transfers.

The user commands are all based on a function library which can be used by any C program to make one's own network connections. Functions exist for connecting, disconnecting, transferring files, reading, writing and signalling along the network. The net looks like a UNIX device driver: stty's are used to manipulate connections to other hosts. A connection is specified by 4 bytes (local-host, local-socket, foreign-host, foreign-socket; sockets are ARPA-style sockets). Many connections are allowed to the same host/socket pair as long as the 4-byte name is unique. This vastly simplifies the establishment of connections.

The whole package is available free as part of the Purdue distribution tape.

Speaker 33, at 11:46   **Networking at NYIT**

Bill Lindemann
Computer Graphics Lab
P.O. Box 170
Old Westbury, NY 11568

Bill spoke about the networking at NYIT, where an 11/35 is used as a front end for most of the terminals. Connections between machines are via DR-11C's which makes it seem as though there are serial lines. As Tom Duff mentioned (see Speaker #21), there are lots of machines to be networked.

Speaker 34, at 11:53   **RT/EMT: an RT-11 emulator on UNIX**

Mike Tilson
Human Computing Resources Corporation
10 St. Mary Street
Toronto, Canada M4Y 1P9
(416) 922-1937

Mike Tilson spoke about HCR's RT-11 emulator. This system actually does two separate things: run RT-11 binaries unchanged on UNIX; and run an RT-11 command interpreter within UNIX.

There is a need for such software because: (a) UNIX provides a good environment for the development of RT-11 software; (b) RT-11 applications programs can be brought up very quickly under UNIX; (c) there are always users who, for reasons of laziness or otherwise, do not want a change in their "operating system" at the surface level.

RT/EMT system overhead is comparable to RT-11. The features of the V3B SJ monitor are supported, and the system runs in user mode, so that minimal or no changes to the UNIX operating system are required. The RT-11 file system is simulated with UNIX directories containing ordinary UNIX files, so UNIX file manipulation programs may be used usefully.

The emt command will run any RT-11 binary, such as MACRO, LINK, LIBR, FORTRAN, PATCH, OMSI, PASCAL, EDIT, or TECO (with the V52 screen). The command rtcom puts

you into the RT-11 world with the exception that the keyboard-editing conventions of UNIX rather than of RT-11 are supported. (Since many UNIX installations now use the DEC conventions anyway, this is hardly a problem.)

The emulator works by loading itself into core in a large user area and copying itself to the top. It takes about 7 Kwords of memory in 16 Kwords of user space. (If thrashing is a problem, it can easily be reconfigured for a smaller memory size.) Other than the startup overhead, the emulator is comparable to actual RT-11 in speed.

RT/EMT is available with full source and documentation, along with a year's worth of bug reports and updates, for US$1350 to commercial users ($540 for each additional CPU); the price for non-profit educational institutions is US$810 ($315 for each additional CPU). There is a small additional shipping charge.

Speaker 35, at 12:20   **Software Tools Users Group Report**

Dennis Hall
Lawrence Berkeley Lab
University of California
Berkeley, California 94720

Dennis summarized the meeting of the Software Tools Users Group held on June 19, 1979 at the Westbury Hotel, Toronto. Of the 96 attendees, 46 were Software Tools types only and 50 were UNIX people as well. Dennis will be writing up the meeting in a forthcoming issue of ;login:.

LUNCH

FRIDAY AFTERNOON
Session 6: Improvements to the UNIX Operating System
Chair: Tom Duff, NYIT

Speaker 36, at 1:45   **Anyone doing system performance monitoring?**

Phil Poulos
Computer Systems Research Group
University of Toronto
Toronto, Canada M5S 1A1

Phil announced that he is doing some work on systems monitoring. He would like anyone who has done evaluation and/or prediction of UNIX system performance to contact him.

Speaker 37, at 1:46   **USENIX Committees**

Lou Katz (speaker #13) announced that committees have been formed with the following chairmen: Site committee (for next summer's meeting): Ben Woznick, BBN, Cambridge. Agenda of the Boulder meeting: John Donnelly, NCAR, Boulder. Tape distribution: Reidar Bornholdt, Columbia University, New York. The Nominations committee for the USENIX Board of Directors has not been formed yet.

Speaker 38, at 1:47   **11/40 Kernel Multiple Address Space**

Alfred Whaley
Department of Computer Science
University of Illinois U-C
Urbana, Illinois 61801

The UNIX installation at Urbana ran into a shortage of buffer space and lack of room for device table entries. The problem was caused by "too good I/O equipment"; one fast powerful multiplexor which handles 128 DMA full-duplex channels, serial and parallel lines, printers, readers, and machine-machine communication.

The immediate goal was to put more code and data into the kernel than will normally fit. What was developed instead was a package of software which makes it possible to reconfigure the system easily with any space problem. There have been attempts at this sort of thing before, but MERT is "too much and too slow", and the Calgary buffer system lets one move only buffers out: when those are gone, then what?

The multiple addressing system developed works with separate "kernels", each with its own virtual addressing, text, data and BSS. Into "SYS" goes all the usual stuff from the sys directory (sleep, wakeup, system calls, scheduler, etc.); "B" devices are block devices such as disk drivers; "C" devices are character devices such as mem, tty's, and others; "BEGIN" contains the main() routine.

A few programs had to be changed to correspond to the kernel changes, notably ld, which now has a -v flag which combines with the "-o outfile" option to split up the kernel into its various pieces. A change to cpp (C preprocessor) gets around having to modify existing C source to run under the new kernel.

The system seems to have worked so far. It's available on tape from Alfred Whaley.

Speaker 39, at 2:05   **UNIX for 1100 Users??!!!**

Ian Johnstone
Australian Graduate School of Management
University of New South Wales
P.O. Box 1
Kensington 2033, Australia

The University of New South Wales has been running UNIX for four years (since Version 5). They have 15 PDP-11's running UNIX. The particular site which Ian described is an 11/70 with: FP11-C floating point; 640 Kbytes of core; two 84 Mbyte UNIBUS disks (which are Ampex DM9100 and NOT recommended); eight DZ-11 multiplexors; 40 VDU's at 2400 baud; 10 DECwriters; and lots of line printers, card readers, tape drives and so

on. The system currently supports 1100 (yes, eleven hundred) users with 13,000 connect hours per month with general access only on weekdays. Between 7 a.m. and 8 p.m., at least 37 terminals are logged in 50% of the time; 43 terminals 25% of the time. Most of the workload consists of undergraduates using the Berkeley Pascal and EM.

To support this kind of use, two types of changes had to be made: those to accomodate a huge user population; and those to improve response with many simultaneous processes.

Changes to accomodate 1100 users include: a restructured passwd file, in binary rather than ASCII, with extra information stored for each entry; resource control (disk space quotas, process quotas, terminal restrictions, terminal booking, and proper accounting); dump and restor for a large file system; error logging in a file rather than the console; recovery from power fail and parity errors; dcopy, a disk compactor; and printf as a system call.

Changes to improve response include: an unrestricted number of buffers with faster buffer lookup via hashing and a pool of headers for raw I/O; a *fully* optimized DZ-11 driver; better process handling; disk driver optimizing; reduced swap activity; improved scheduling; and lots more.

A comparison between the virgin V6 UNIX and the current UNSW UNIX shows: maximum number of simultaneous users increased from 37 (with intolerable response times) to 44 (with satisfactory interaction); average time to compile and execute a 2-second Pascal program down from 158 seconds to 32; compile and link /unix (on an idle system) down from 10 minutes to 6. The first two results were obtained on a fully active system under normal operating conditions. System monitoring shows that the percentage of CPU time spent in "user" rather than "sys" mode has increased, on average, from 15% to 31%.

The entire UNSW system-improvements package is available on the conference distribution tape. The file /usr/sys/defines.h contains full information about the changes.

Speaker 40, at 2:30   **Another large UNIX system**

George Goble
Electrical Engineering Department
West Lafayette, Indiana 47907

George described the 11/70 at Purdue with a user population of 1400. Last year the system supported 83 at once at the worst time; this fall they expect 60-85 simultaneous users. With that amount of computing to be done, compiles and nroffs will probably be forked off to other machines with Bill Croft's networking (see Speaker #32). Because of the number of users, GID's are not used. The ka5 address is extended to fit in extra buffers, procedures and so on: between BSS and the kernel there is a new area, starting at virtual 120,000. This whole region is less than 64 Kwords. The system has a dual swap device, with a fast primary disk (RS04) and slow secondary swapping. Small programs are, of course, swapped on the fast disk.

There is a new system call, meer(II), which will let one change the priority on another process. The set-gid bit is used (since there are no groups) as a "game" bit — the process runs at rock-bottom priority. There is also a "research" bit, for programs run in background mode which compute only when there is no interactive process running. Power failure recovery has been implemented. Automatic file system recovery is per-

formed by *bootchk*, a program written in 1977 by Mike Tilson at the University of Toronto.

The Purdue people have had a lot of problems with their UNIBUS disks and recommend that one should never put a disk on the UNIBUS. Use the MASS BUS.

The entire collection of system changes, along with lots of other goodies (including the MIT dungeon) is available from George for free. Send him a tape and return postage if you want a copy of the Purdue distribution.

Speaker 41, at 2:50   Office Use of UNIX

Walter Lazear
Air Force Data Services Center
The Pentagon, Room 1D988
Washington, D.C. 20330
(202) 695-6161

The Air Force Data Services Center has four 11/70's for administrative support, handling text processing and similar tasks. The machines are up 22 hours a day, 7 days a week running practically nothing but nroff, troff and editing. Each 11/70 runs V6 with 40 simultaneous users. In addition, there is an 11/35 for the programmers to play with, which has compilers and so on. All machines are equipped with auto-dialers, so they can call each other or anyone else.

Walter described modifications to UNIX made at AFDSC. One such mod is the proper implementation of exit codes, including a specific code for "bad format". When the shell receives this exit code from a process, it looks up the correct usage in the manual section and informs the user. Furthermore, a consistent user interface has been developed for argument formats in all programs.

Other changes include: a read-only root file system, separate from the user file system; a system call telling you what machine you're running on; and a strategy for solving the problem of bad disk blocks, particularly in the swap area. The last 48 blocks of each file system are reserved as "alternates", and a table in the super block (the last 48 words) holds the bad block list for that file system. Mkfs has been modified to read and write every block in the file system, note all the bad blocks, and only then mount it. As a result, mkfs takes about 45 minutes to make a 56 Kbyte file system; however, once made, file systems run much better. The critical area is reduced to only two blocks (boot and the super block), and even programs which know about the internals of the file system (like df) will still work. Since installing the change, a total of 18 system-months have been logged on three systems without disk error. The overhead for the process is less than 1%.

The collection of AFDSC mods is available from Walter on a distribution tape.

coffee

Speaker 42, at 3:50   Implementing C and UNIX more efficiently

Carl D. Howe
Bolt, Beranek & Newman, Inc.
10 Moulton Street
Cambridge, Mass. 02139
(617) 491-1850 x3642

Carl described the environment at BBN, where they have 8 DEC-10's going full blast (and work for lots more). They have an 11/70 running UNIX with 150 users. There is, however, a demand for a machine with the performance of the 11/70 (running UNIX) at much less cost.

The BBN Computer Company (a subsidiary of BBN) has designed the Microprogrammable Building Block (MBB), a fast microprogrammable processor to be used as the basis for special purpose computer systems. This processor features a 135 nanosecond microcycle time and up to 16K of 32-bit microcode memory words. The machine also has 1024 fast hardware registers. Its architecture makes emulation of already-existing instruction sets easy and efficient. Furthermore, it is inexpensive to build and easy to service.

The MBB has been built and is being used for other projects. It emulates the Honeywell 316 one and one-half times *faster* than the 316 runs its own code.

The MBB has a *macrocode* which is distinct from its microcode. The macrocode is an intermediate code produced by the compiler. It is *not* designed as a users' assembler and hence is often ugly but efficient. The objective was to optimize the instruction set in the direction of efficiency at the expense of aesthetics.

In constructing a C compiler for the MBB it was desired to avoid some of the problems with the PDP-11, such as the 16-bit address limit, the limit of three hardware registers available to the user, the slow speed of subroutine calls and the inefficiency of local variable addressing. These problems have, it seems, all been solved. With 20-bit words, larger processes may be run on the machine. The abundance of hardware registers makes register saving unnecessary (except with large amounts of recursion, in which case 64 are saved at once). The macrocode can call the microcode to do certain things; there is flexibility in deciding what will be implemented in microcode. And the microcode has an instruction set with many different addressing modes, so that doubles and longs can be dealt with directly.

The assembler for the MBB was written in 3 days (using yacc). The compiler is being written and compiled with the phototypesetter-version of cc. The microcode has been written and the hardware already exists although it is not yet generally available.

Once C is working, BBN will begin implementing UNIX on the MBB. UNIX will feature: a 20-bit MBB processor; 128K 20-bit words of memory; two disk drives of 40 to 300 megabytes; one 8-line serial multiplexor for terminals; and a compute network interface. It is expected to be running by the end of the summer; disk interfaces and memory management have not been designed yet. While nothing can be promised as yet, it is expected that BBN will be marketing the MBB under UNIX by the summer of 1980, and that it will give the performance of an 11/70 for the cost of an 11/34.

Speaker 43, at 4:10

UNIX on a UNIVAC V77-800

Harold Pierson
RLG Associates, Inc.
11250 Roger Bacon Drive, Suite 16
Reston, Virginia 22090
(703) 471-1106

Harold Pierson described UNIX v77 on the Univac as "an experience in portability". RLG has been doing the project for Sperry Univac, who took over Varian and are producing the "Varian V77" as the "Univac V77". The writing of the C compiler was subcontracted to Cassandra Inc. (Bob McClure). UNIX is now up and running most of PWB on the V77.

Harold discussed some of the problems resulting from the fact that the V77 is a word-addressed machine. Assignment conversions, parameter passing and word/byte relocation in the loader all presented a problem. The solution implemented was to pass always by byte pointer when calling subroutines. Within routines, word pointers are used. The loader was modified to cope with this.

The were other assorted incompatibilities. The final result was code that is larger by 20-30% on a machine which is much more powerful and faster than the PDP-11. Additionally, because of 512-word paging, there are 16 address spaces in core; instead of just the kernel, the supervisor and the user you can actually have 15 users running in core.

At present UNIX (the real thing, not a look-alike) is running multi-user on the V77-800. RLG is handling the program development and administrative software; Sperry Univac will be marketing it in the future. Other developments to follow are a version for the v77-800 (with separate I and D space), multiple address spaces, scatter page swapping and micro-coded assists.

Speaker 44, at 4:35

An Accounting System for UNIX

Robert N. Jesse
Johns Hopkins University
Electrical Engineering Dept.
Barton Hall
Baltimore, Maryland 21218

The 11/45 at JHU is being used by students, which automatically creates a hostile environment. All the things that one can normally get away with on UNIX must be monitored and controlled.

The /etc/passwd file has been left intact, but a parallel file, /etc/uif, contains more information for each user: the encrypted password; time and tty of last login; cpu and connect time used and quotas; block-I/O and line-printer-page count and quotas; login terminal restrictions; and special permissions and options.

Monitoring of system usage is done by a new program, init2, which receives the termination statuses of all programs run by the used on a pipe from init. The information is passed with a new system call, waitinfo(), which provides the proc and user structures of the deceased process. When the shell exits, init calls quota, which will not let the user log out until he has reduced his disk consumption below his quota. (If the user is on a phone line and hangs up, he will not be permitted to log in again by phone until he

has discussed his problem with the system administrators.) Another system call written for init2 is getppid(II), which returns the PID of the parent. The program qcheck is run periodically to check for people owning files in others' directories.

When the distribution of this system is ready, it will be available to educational UNIX licensees only. Those wishing to obtain it must sign a non-distribution agreement (the JHU people want all copies to come from them).

Speaker 45, at 4:55

A High Performance UNIX System

Mike Muuss
Johns Hopkins University
Electrical Engineering Department
Barton Hall
Baltimore, Md. 21218

Because it was late, Mike spoke very briefly about the changes made at JHU to improve performance. "Performance" in this context is not just good response time; it includes:

(a) excellent response to small interactive processes, while keeping the CPU as full utilized as possible;
(b) good security in a hostile user environment;
(c) detection and avoidance of certain hardware failures;
(d) good data integrity, especially in a hostile hardware environment;
(e) a more pleasant user interface; and
(f) accountability and restrictions on resource consumption.

The JHU people have divided the solution up into four areas: (1) resource measurement and accounting; (2) teletype driver; (3) security and integrity; and (4) performance mods.

Resource measurement. See Robert Jesse's talk (Speaker #44) for a description of the accounting system. The file /dev/data contains dynamically modified information about processes running, system load and performance, and resources.

Teletype driver. Lots of enhancements increasing performance; others adding a better user interface: ctrl-r to retype the buffer; page mode; stall mode; character rubout via BS-SP-BS; ctrl-t to print status of process being waited for.

Security and integrity. Special files and SET[UG]ID operate only on the root file system; front panel interference protection (!); /dev/error and /etc/logerr for logging and recording history of non-fatal errors, along with interpretative software: memory test and lockout at boot time and dynamic memory testing; a stack limit register; use of IDB - kernel runs a true split I and D space, with the I-space write protected; fixing of assorted errors and bugs in the system source.

Performance mods. Process dispatching has been fixed up by implementing full process queues. The result has been much faster reponse time for editors and other small CPU usage programs. ($\frac{1}{2}$-second typical reponse, with 10-15 users on the 11/45). Swapping increases, however, so it is advisable to have an RK05 dedicated to swapping. In the future, I/O will be integrated into the dispatching priority. The priority evaluation done now keeps track of how long the process was in core before and takes into account the size of the process.

Some fatal bugs in the system source were also found, but Mike could not discuss them publicly because of the presence of "aliens" (people not bound by a Western Electric non-disclosure agreement).

Speaker 46, at 5:15   **UNIX on the IBM 370**

Steve Bellovin
University of North Carolina
New West Hall 035 A
Chapel Hill, North Carolina 27514

Steve spoke about the Triangle Universities Computing Center (TUCC) group, which has been running TSO for six years and would now like to move to a time-sharing system. The plan is to move the command level of UNIX to an IBM S/370-165/II, and thence to the Amdahl 470. The reason these machines have been chosen is that their hardware and hardware service are felt to be far more reliable than DEC.

The 370 UNIX will be CRT-terminal oriented. It will be an EBCDIC system (ugh!). Assorted stuff will be imported from other systems. It will be implemented using VM-370; the goal is to have 100 simultaneous users, each with a virtual machine. The UNIX file system, pipes and shell will be kept intact.

It is hoped that in a year's time the system will be operational and capable of handling 20-30 simultaneous users. By two years from now it should be on the market. Steve has no idea yet as to what the price will be, or whether it will be available cheaply to educational institutions.

Does anyone have a DX-11 driver for Steve?

SATURDAY MORNING
Session 7: Educational uses, and what's happening at ?....
Chair: Sandra Wright, DCIEM

Attendance was low at Saturday's session (about 35 instead of 350). These notes were taken by Sandy Wright and expanded by Dave Sherman.

Speaker 47, at 9:05   **UNIX in the Undergraduate Lab Environment**

Don Schertz
Bradley University
Dept. of Electrical Engineering
Peoria, Illinois 61625

UNIX at Bradley is runnning on an 11/40, where half of the connect time is spent on software development for a microprocessor. RJE into the university's CYBER is planned in the near future. They are running Mini-UNIX from the Johns Hopkins University, which is a good package. Communications to micros is being worked on. Currently RS-232 loops are used, but there will be a move to the 8086 on parallel ports. With a variety of cross assemblers available, the micros are downloaded with binaries.

Speaker 48, at 9:15   **UNIX in a large educational environment**

Mike O'Dell
University of Oklahoma
Engineering Computer Network
Engineering Center
Norman, Oklahoma 73019

The University of Oklahoma's 11/70 has 384K of core with a floating-point processor and RM03's on separate controllers; four DH-11's (8 dialup lines), and a user population of 500 to 700. The maximum number of users supported simultaneously is 37. They use the Purdue tty driver with some mods for weird terminals.

Work is being done on RJE to the IBM 370/158; they are trying to get the PWB RJE software working.

Speaker 49, at 9:30   **QED, or The Little Ed That Grew**

Robert Pike
Caltech 220-47
Pasadena California
USA 91125

David Tilbrook
Bell-Northern Systems Research
247 Brunswick Avenue
Toronto, Canada M6S 2M6

CSRG at the University of Toronto has had ed hacks for a long time. Years ago, Tom Duff and Rob Pike added assorted useful features; these were expanded and rewritten recently by Hugh Redelmeier. Ed at U of T now has the following features: a single error character following the '?' indicating what type of error was found; '*' as an easier way of typing 1.$ (or N* for N,$); '&' as a simple form of page-oriented addressing; file saving upon a hangup signal (in "filename:sav"); extensions to regular expressions for easier matching; s2/xxx/yyy/ for the second occurrence in a line (and so on); 'cd file' for changing the working directory; 'x' command for character editing; 'j' to join lines; 'u' to undo the last substitution; and others.

Qed was originally written by Tom Duff by taking the code for ed and adding multiple buffers, number registers and assorted other goodies. Improvements by Hugh Redelmeier, David Tilbrook and Rob Pike have taken place over the intervening four years. Although more enhancements are still in progress, qed is already a powerful editing tool.

The principal features of qed are multiple file/multiple buffer editing, macro capability and an enhanced interface to the shell. For experienced programmers, these features can greatly simplify difficult or repetitive text editing tasks, and make it easy to develop special-purpose editors that have knowledge about the structure of the text they are working with.

As much as possible, qed is a rigorous superset of the version of ed running at U of T. The couple of minor differences are considered to be improvements over the original ed, but it was decided to leave ed's behaviour as it was to avoid incompatibilities with eds running elsewhere (specifically the PWB (Bench) version).

The main extension in qed is the presence of 56 buffers, labeled by upper and lower

case alphabetics and four other characters which are reserved for several purposes. All normal ed commands work within a buffer, so that internal to a particular buffer the user sees what appears to be a regular editor. The only exceptions to this are the move and copy commands, which allow inter-buffer transfers, and, of course, the command to change the current buffer.

At any time when input is expected, be it command or append input, the contents of a buffer may be inserted using a simple escape sequence. As well, qed provides registers which are accessed in a similar manner, but which have some commands to deal with them directly rather than by changing the current buffer. The user, at startup, can specify the name of a file which is to be read into a reserved buffer and executed before reading commands from the terminal, so that he can preset registers and buffers to contain useful commands and text.

For programming purposes, qed also has simple control structures and message printing capabilities, so that it can be used as a rather simple programming language. The implementers of qed claim that their version is considerably better human engineered than previous versions of qed on which it is based.

A tape with source and documentation for qed, U of T's ed, and a few other useful programs, may be obtained from Robert Pike. Please send a blank tape, address label and return postage.

coffee

Speaker 50, at 10:50   **News from the U.K.**

R.P.A. Collinson
University of Kent
Canterbury CT2 7NF
Kent, England

The are 15-20 UNIX sites in the United Kingdom, all with educational licenses. Most of them are 11/40's, although there are three 45's and one 70. They are in touch with European UNIX sites in Holland and France. Only Glasgow has tape drives — RK05's are used for the most part, which has the side effect of making distributions more selective and carefully thought out.

As there are only perhaps five system hackers in the whole of Great Britain, most locations stick to the conventional stuff, using "Standard C" (whatever that is), and keeping any changes to system source code well documented.

Languages being developed in Britain include: BCPL (two different versions of this portable high-level predecessor to C); Modula (at the University of York: Ian Cottam); PDP11 (Sussex: Steve Hardy); and micro assemblers in various places. An implementation of Algol 68 — a major project — is underway at St. Andrews.

The UK UNIX User Group has been in operation for two years. It is a DEC SIG; it publishes a (theoretically) quarterly newsletter. Glasgow acts as a software distribution centre. The users' meetings draw about 50 people. UK/UUG is prepared to exchange tapes with USENIX.

Speaker 51, at 11:05   **What's happening at BBN**

Carl D. Howe
Bolt, Beranek & Newman, Inc.
10 Moulton Street
Cambridge, Mass. 02139
(617) 491-1850 x3842

Carl briefly mentioned a number of the projects underway at BBN: the new C machine (see speaker #42); a C compiler for the Z8000 and the DEC-10; manual revisions; extended documentation; non-blocking I/O; ports, networking; a new Rand editor — for CRT's with cursor addressing and shared text; word operations user configurable; cross-net debugging; screen managers under investigation; ARPAnet work; work on high bandwidth local networking is starting.

The BBN UNIX mods distribution tape costs $300.

Speaker 52, at 11:15   **Tuning PWB UNIX**

George Pajari
GTE Automatic Electric (Canada) ltd.
100 Strowger Blvd.
Brockville, Ontario, Canada K6V 5W8

The RJE modification under PWB needs rewriting. Kernel modifications have also been made at GTE: a query feature which gives utilization of tables; and gmon, which keeps track of high- and low-water marks in tables. Knowledge about these watermarks allows sensible settings of table sizes, and also prevents some crashes by detecting table overflows.

This software might be available. Contact Paul Hart at GTE (address above).

Speaker 53, at 11:25   **Screen editors**

Mark Pearson
Yourdon Inc.
1133 Avenue of the Americas
New York, NY 10036

Yourdon has a screen editor which works as a front end to ed. It supports dumb terminals. The backslash character is not special. The cursor can be moved a word at a time. It should be able to front onto the Toronto qed, and it was agreed by Dave Tilbrook and Mark Pearson to try and do so.